Large-scale electromagnetic field analyses of coils wound with coated conductors using a current-vector-potential formulation with a thin-strip approximation

Mifune, Takeshi; Tominaga, Naoki; Sogabe, Yusuke; Mizobata, Yudai; Yasunaga, Masahiro; Ida, Akihiro; Iwashita, Takeshi; Amemiya, Naoyuki

Large-scale electromagnetic field analyses of coils wound with coated conductors using a current-vector-potential formulation with a thin-strip approximation

Takashi Mifune[1], Naoki Tominaga[1], Yusuke Sogabe[1], Yudai Mizobata[1], Masahiro Yasunaga[1], Akihiro Ida[2], Takeshi Iwashita[3] and Naoyuki Amemiya[1,*]

[1] Department of Electrical Engineering, Graduate School of Engineering, Kyoto University, Kyoto-Daigaku-Katsura, Nishikyo, Kyoto 615-8510, Japan
[2] Supercomputing Research Division, Information Technology Center, The University of Tokyo, Yayoi, Bunkyo, Tokyo 113-8658, Japan
[3] Information Initiative Center, Hokkaido University, W5 N11, Kita, Sapporo 060-0811, Japan

*Corresponding author: amemiya.naoyuki.6a@kyoto-u.ac.jp

Abstract

We developed a novel software for large-scale electromagnetic field analyses of coils wound with coated conductors based on current-vector-potential formulation with thin-strip approximation. Although this formulation was effective for obtaining the precise solutions of the electromagnetic field, the strong nonlinear property of superconducting materials frequently led to highly ill-conditioned linear systems of equations, which were difficult to solve efficiently. Moreover, the memory consumption and computation time required for the analyses rapidly increased with the size of the analysis due to dense matrix operations. In our software, the first difficulty was addressed by a novel preconditioning technique based on algebraic multigrid method. Algebraic multigrid preconditioning enabled us to efficiently and stably solve the ill-conditioned linear systems of equations encountered in our analyses. It also improved the robustness of the analyses containing multifilament-coated conductors. As regards the second difficulty, the hierarchical matrices representation drastically reduced the memory consumption related to the dense matrices, as well as computation time. Meanwhile, our implementation of the hierarchical matrices representation was quite compatible with parallel computations on distributed memory computers. Finally, we presented some practical examples of large-scale analyses, which became possible by using the new software. For instance, the analysis of a cosine-theta dipole magnet whose number of degrees of freedom was more than 1.5 million was successfully completed in 78 h by 56 parallel processes and with a total memory consumption of 177 GB.

Keywords: ac loss, algebraic multigrid, coated conductor, electromagnetic field analysis, hierarchical matrix, Newton–Raphson, screening current, shielding current

1. Introduction

Electromagnetic field analysis is a powerful tool for studying ac losses and shielding-current-induced fields (SCIF) in superconducting coils [1]–[13]. Meanwhile, the ac losses and SCIF of superconductors are studied using eddy-current analyses. However, the strongly-nonlinear resistivity of superconductors is complex in comparison with the eddy current analyses of normal conductors, in which their conductivities are usually constant. Furthermore, the increased aspect ratios of the cross-sections of coated conductors pose some problems when their thin superconductor layers are analyzed using three-dimensional finite element method. This is because elements with low aspect ratios are usually preferred. Thus, this issue can be avoided using thin-strip approximation and formulation based on current vector potential [1]–[3], which will be explained in the following section. This approach reduces the number of degrees of freedom (DOFs) used in the modeling of an entire coil wound with coated conductors, since the mesh does not extend along the thickness of the superconductor layer.

Current-vector-potential formulation with thin-strip approximation substantially reduces the cost of mesh generation. However, it involves two major difficulties that can hinder the robustness and scalability of analysis. Firstly, the formulation may lead to a highly ill-conditioned linear system of equations, especially when the electrical conductivity of the materials has a strong nonlinear property. This becomes even more dramatic when the analysis model contains materials whose conductivities vary incomparably, such as multifilament coated conductors [14], such that the simulation becomes unstable in some circumstances. Secondly, the computational kernel of the analysis has an $O(N^2)$ complexity since large-size dense matrices are derived from the formulation, thereby strongly limiting the scalability of analyses. Although fast multipole method (FMM) [15], [16] can be used to reduce the cost of dense matrix computation [11]–[13], FMM is difficult to implement and its applications are difficult to generalize.

In this study, we aim to overcome the above difficulties using novel numerical methods and state-of-the-art techniques. First, we develop a new algebraic multigrid (AMG) preconditioning technique for the linear systems of equations in our analyses. This is required due to the occurrence of a serious deterioration of convergence of the linear iteration when Newton–Raphson method is applied to the nonlinear system of equations derived from the current-vector-potential formulation. The details of AMG preconditioning and related numerical examinations are given in Section 3. Furthermore, we investigate the effect of AMG preconditioning on improving the robustness of the analyses of models containing multifilament coated conductors in Section 4. In Section 5, we discuss an application of the hierarchical matrices (H-matrices) representation that can significantly improve the scalability of our analyses and has flexibility to deal with various analysis models. Finally, some practical examples of analyses with large numbers of DOFs are presented in Section 6.

2. Time marching analysis with nonlinear and linear iterations

2.1. *Current-vector-potential formulation with thin-strip approximation* [2], [17]

A current vector potential $\boldsymbol{T}$ was formulated using current density $\boldsymbol{J}$ as follows:

$$\boldsymbol{J} = \nabla \times \boldsymbol{T}. \tag{1}$$

Meanwhile, thin-strip approximation was used due to the thin-film structure of the coated conductor. The conductive region was regarded as an infinitely thin film such that only the current density component that is tangential to the analyzed layer was considered [18]. Thus, the governing equation derived from Faraday and Biot–Savart laws is given by [19]:

$$\nabla \times \left(\frac{1}{\sigma_s}\nabla \times \boldsymbol{n}T\right)\cdot \boldsymbol{n} + \frac{\partial}{\partial t}\left(\frac{\mu_0 t_s}{4\pi}\int_{S'}\frac{(\nabla\times \boldsymbol{n}'T')\times \boldsymbol{r}\cdot \boldsymbol{n}}{r^3}\mathrm{d}S'\right) + \frac{\partial B_{\mathrm{ext}}\cdot \boldsymbol{n}}{\partial t} = 0. \tag{2}$$

where T and T' are the current-vector-potential components normal to the analyzed layer at the field point (where the field or potential is calculated) and the source point (where a current flows to generate the magnetic field at the field point), respectively; $\boldsymbol{n}$ and $\boldsymbol{n}'$ are the normal vectors of the analyzed layer at the field point and the source point, respectively; $\boldsymbol{r}$ is the vector from the source point to the field point; σ_s is the equivalent conductivity of superconductor; and S' is the entire area (wide face) of the analyzed layer. S' consists of triangular elements and is curved in a coated conductor wound into a coil. The details of the model are given in [2], [17].

The Galerkin and backward finite difference methods [20], [21] lead to the following system of equations:

$$\left(A(\sigma_s) + \frac{\mu_0 t_s}{4\pi\Delta t}B\right)\boldsymbol{T}_{[m]} = \frac{\mu_0 t_s}{4\pi\Delta t}B\boldsymbol{T}_{[m-1]} + \boldsymbol{C}. \tag{3}$$

where matrices $A(\sigma_s)$, B, and vector $\boldsymbol{C}$ stem from the first, second, and third terms of the left-hand side of Equation (2), respectively; Vector $\boldsymbol{T}_{[m]}$ represents the discretized current-vector-potential at time step m, and Δt represents the time interval between time steps m and $m-1$. Meanwhile, B is a dense matrix due to the integral operation in Equation (2), while A is a sparse matrix. Equation (3) is a nonlinear system of equations due to the nonlinear property of the equivalent conductivity, which is referred to below.

2.2. Constitutive relation

The constitutive relation was based on an extended Ohm's law in which the constant conductivity is replaced with the equivalent conductivity of superconductor σ_s:

$$J = \sigma_s E, \tag{4}$$

where J and E are current density and electric field, respectively. The superconducting property was obtained using E–J power law relation [1], [22]:

$$E = E_0 \left(\frac{J}{J_c(B_\perp)}\right)^n, \tag{5}$$

where E_0 is 10^{-4} V m^{-1}, and J_c is the critical current density which is solely determined by the magnetic field component normal to the superconductor layer $B_\perp$, using Kim's model. J_c is given by:

$$J_c(B_\perp) = J_{c0} \frac{B_0}{B_0 + |B_\perp|}, \tag{6}$$

where J_{c0} is the critical current density at zero magnetic field and B_0 is a constant [23]. Hence, the equivalent conductivity of the superconductor $\sigma_s(\boldsymbol{n}T)$ is given as:

$$\sigma_s(\boldsymbol{n}T) = \frac{J}{E} = \frac{J_c(B_\perp)}{E_0}\left(\frac{J_c(B_\perp)}{J}\right)^{n-1} = \frac{J_c(B_\perp)}{E_0}\left(\frac{J_c(B_\perp)}{|\nabla\times\boldsymbol{n}T|}\right)^{n-1}. \tag{7}$$

For some situations, we may set $B_0 = \infty$, i.e.,

$$J_c(B_\perp) = J_{c0}, \tag{8}$$

for simplifying the discussion and examination.

2.3. Computation process of entire analysis

The computation process of the entire time-marching analysis based on Equation (3) is described using triple-nested loops, as shown in Figure 1. The outermost loop corresponds to the time step iteration, where m is assigned to the loop variable. At each time step, a nonlinear system of equations with large numbers of DOFs need to be solved. However, most methods of solving nonlinear equations divide the problem into a series of linear systems of equations with the same number of DOFs as the original problem. Meanwhile, solving the large linear systems of equations by direct methods may be too expensive. Thus, linear iterative solvers such as BiCGSTAB method [24] is a feasible approach. Hence, the middle and innermost loops of our analyses correspond to the nonlinear and linear iterations, respectively.

It is important to improve the convergence property of both nonlinear and linear solvers to perform the entire analysis efficiently and stably. In this study, we adopt the preconditioned BiCGSTAB method as a linear solver and devise a new preconditioning method based on AMG method [24], [25]. The motivation of the new approach is related to the linear solution process and our choice of the nonlinear solver, as explained in the following section.

Furthermore, the $O(N^2)$ computational complexity needs to be overcome to execute analyses with a large number of DOFs, due to the dense property of matrix B. For example, an analysis with one million DOFs requires at least 8 TB memory capacity, if the dense matrix is stored in a naïve manner. This problem is addressed using H-matrices representation, as described in Section 5.

3. Algebraic multigrid preconditioning in Newton–Raphson method

3.1. Convergence deterioration of linear solver applied in Newton–Raphson step

The Newton–Raphson method is one of the most widely used iterative solver for large nonlinear system of equations such as Equation (3). Generally, Newton–Raphson iteration converges much more rapidly than the more primitive solvers that require no computation of derivatives, such as the successive substitution method. However, some authors adopted successive substitution method as the nonlinear solver for Equation (3) in their works [2], [3], due to the following reason:

The difference between successive substitution and Newton–Raphson formulation is due to the determination of the values associated with the electrical conductivity. Moreover, successive substitution method applied in Equation (3) results in a linear system of equations with the following coefficient matrix:

$$K_{\mathrm{SS}} = A\left(\sigma_{\mathrm{s}}\left(\boldsymbol{n}T_{[m]}^{l-1}\right)\right) + \frac{\mu_0 t_{\mathrm{s}}}{4\pi\Delta t} B \tag{9}$$

where l indicates that the subscripted value is computed from the solution of the l-th nonlinear iteration step. On the other hand, the coefficient matrix based on Newton–Raphson method is given as:

$$K_{\mathrm{NR}} = A\left(\left.\frac{\partial \boldsymbol{E}}{\partial \boldsymbol{J}}\right|_{T=T_{[m]}^{l-1}}^{-1}\right) + \frac{\mu_0 t_{\mathrm{s}}}{4\pi\Delta t} B \tag{10}$$

where $B_0 = \infty$ for simplicity. Otherwise, the evaluation of $\partial \boldsymbol{B}/\partial \boldsymbol{J}$ is also required since $\boldsymbol{E}$ depends on $\boldsymbol{B}$ as well as $\boldsymbol{J}$. However, the solution of the linear systems of equations with coefficient matrix K_{NR} was frequently computational-intensive in our past analyses, although Newton–Raphson iterations converged more rapidly than successive substitution iterations. Therefore, successive substitution method was adopted in our past works.

The poor performance of the linear solver is due to the following reason: The precise expression of matrix $A(\sigma)$ is given by:

$$[A(\sigma)]_{ij} = \int \nabla \times (\boldsymbol{n} w_i) \cdot \sigma^{-1} \nabla \times (\boldsymbol{n} w_j)\, \mathrm{d}S, \tag{11}$$

where w_i denotes the i-th shape function used in the discretization of T. The matrix is sparse and quite similar to the matrices based on standard finite element discretization of the two-dimensional Laplace equation. Thus, the condition number of $A(\sigma)$ may become very large when the problem size is large. In our application, $|\partial \boldsymbol{E}/\partial \boldsymbol{J}|^{-1}$ can be much smaller than σ_{S} since the equivalent conductivity of the superconductor is given by Equation (7). A situation may cause $A(\sigma)$ to become dominant in Equation (10) and consequently lead to a linear system of equations that is highly ill-conditioned. In fact, the convergence of the linear solvers in Newton–Raphson iteration substantially deteriorated in our past analyses such that the total computational performance was adversely affected.

3.2. Algebraic multigrid preconditioning

If the convergence deterioration of the linear solver used in Newton–Raphson iteration is due to the ill-conditioned matrix $A(\sigma)$, a promising remedy is to adopt a solver/preconditioner that is well suited for this particular matrix. The inversion of the matrix given by Equation (11) is one of the most fundamental problems in numerical analysis for which several fast solvers exist. However, AMG method was adopted since it can handle the problem effectively. The method requires only the coefficient matrix and the right-hand side vector as the input to itself. Meanwhile, it does not require any other problem-dependent information. Hence, various analyses of different shapes of coils can be easily performed. Furthermore, the method can adaptively handle inhomogeneous material characteristics which result from the nonlinear property of the electrical conductivity in our analyses. Moreover, it is quite effective for large-scale analyses [24]. Here, the AMG method described in [25] was used in a preconditioner of the BiCGSTAB solver.

Although the coefficient matrix in standard multigrid applications is assumed to be sparse, K_{NR} is a dense matrix since B is dense. Thus, it is not feasible to directly apply the AMG algorithm to this dense matrix from the viewpoint of computational efficiency. Therefore, we have devised a sparse matrix approximation of the coefficient matrix. If $A(\sigma)$ is sparse and is dominant in K_{NR}, it should be a fair approximation to filter K_{NR} based on the sparse pattern of $A(\sigma)$. In other words, the (i, j) entry of K_{NR} is dropped if the corresponding entry of $A(\sigma)$ is zero. It should be noted that this approximation affects only the input matrix to the AMG algorithm. The linear system of equations solved by BiCGSTAB method and the solution obtained remain unchanged.

3.3. Numerical experiments

Table 1 shows the details of computers used in the numerical experiments in this study. The experiments in this section were executed on host kf01. Intel Fortran Compiler 2018 with –O3 option and Intel MPI (Message Passing Interface) library 2018 were used in all the experiments performed in this study. We examined the effect of AMG preconditioning on a test model of stacked pancake coils, as illustrated in

Figures 2 and 3. The specifications of the model are listed in Table 2. In these analyses, the analyzed object was the top single pancake coil of the stack. The remaining pancake coils were approximated with one line current at the center of each coated conductor composing the pancake coil. The magnetic flux density in the analyzed pancake coil generated by the nonanalyzed pancake coils was then calculated using Biot–Savart's law and treated as B_{ext} in Equation (2). The current in all the pancake coils was ramped up to 50 A in 10 s with a time step of 0.5 s; namely, we have 20 time steps in the test numerical experiments. Most of time-consuming parts of the computations, e.g., matrix-vector multiplications, can be parallelized in a straightforward manner, except for AMG preconditioning. Meanwhile, the proportion of AMG preconditioning to the total computation time was quite small due to the sparse matrix approximation. In our present calculations, AMG preconditioning was sequentially executed.

Figure 4 shows the number of nonlinear iteration and computation time in each time step when BiCGSTAB solver with no preconditioner was used. Although Newton–Raphson method substantially accelerated the convergence of the nonlinear iteration, it failed to reduce the total computation time. This effect is illustrated by Figure 5, which shows the convergence profiles of BiCGSTAB iteration at the first nonlinear iteration step in the 10th time step. When Newton–Raphson method was used, the convergence property of BiCGSTAB method with no preconditioning was significantly deteriorated. These results support the discussion in Section 3.1. As is also shown in Figure 5, the newly developed AMG preconditioning achieved the remarkable convergence acceleration. Furthermore, the nonlinear and linear solvers were stably convergent in the other time steps.

Figure 6(a) shows the comparison of the total computation time, where the computation time of the AMG preconditioning was doubly reduced. Figure 6(b) shows the result obtained when the number of turns of coils is increased to 20, in which the advantage of AMG preconditioning is increased. The effectiveness of the AMG preconditioning will further increase in larger-size practical analyses, as is generally observed in multigrid applications.

4. Effect of algebraic multigrid preconditioning on analyses of multifilament coated conductor

4.1. Problems in analysis containing multifilament coated conductor and algebraic multigrid preconditioning

In this section, we examine the effect of AMG preconditioning on the analyses of multifilament coated conductor. Figure 7 shows a sample model involving a multifilament coated conductor. The magnitude of electrical conductivity of the normal conductor regions were set to be extremely low in comparison with the superconductor regions. Since low conductivities increase the contribution of $A(\sigma)$ in Equation (9)/(10), the impact of the convergence deterioration discussed in the previous section is much more profound.

The problem addressed in this section can be solved using AMG preconditioning since it is the same problem encountered in the previous section. Moreover, the use of AMG preconditioning may drastically improve the situation since $|\partial \boldsymbol{E}/\partial \boldsymbol{J}|^{-1}$ or σ_{s} can be much lower than those in Section 3 in the analyses of the multifilament-coated conductor.

When the input matrix is constructed for the AMG preconditioner as in Section 3, the matrix entries that stem from normal conductor regions have much larger values than the others. Although the computational performance may be further improved based on this fact, we have adopted the sparse matrix approximation described in Section 3 due to its convenience of implementation.

4.2. Numerical experiments

The numerical experiments described in this section were executed on host kf01. The effectiveness of AMG preconditioning was examined in a test model, as shown in Figure 7. Table 3 lists the specification of the test model. The analyzed object is the same coil described in Section 3.3 except that it is wound with multifilament coated conductor. The current profile and time step were also the same with those described in Section 3.3.

Figure 8 shows the convergence profiles of BiCGSTAB iteration with and without AMG preconditioning, at the first nonlinear iteration step in the first time step. The existence of the normal conductor layer significantly declines the convergence of the BiCGSTAB method with no preconditioner such that the simulation cannot proceed. However, this problem is solved by AMG preconditioning. Moreover, Figure 9 shows a fluctuation in the number of nonlinear iterations when AMG preconditioning is used. Thus, Newton–Raphson iterations stably converge and the time-marching analysis of multifilament coated conductor can be successfully performed using the new preconditioning technique.

5. Use of hierarchical matrices for reduction of computational complexity due to dense matrices

5.1. Hierarchical matrices and its application to our analyses

In the previous two sections, we demonstrated that the computational performance of our analyses, especially with respect to computation time, can be significantly improved using AMG preconditioning. However, the remaining performance bottleneck of our analyses is the $O(N^2)$ computational complexity due to the large-size dense matrices. Actually, the maximum size of the problem that can be handled in our computational environments is primarily limited by the memory consumption of the dense matrices. In addition, the most time-consuming part of our analyses as regards the computation time is the matrix-vector multiplications with respect to the coefficient matrices, which are located in the innermost part of the triple-nested loops and are executed twice per BiCGSTAB step.

The dense property of the coefficient matrix stems from the dense matrix B, which can be regarded as a certain type of discretized integral operator. Thus, we aim to handle these matrices efficiently, using H-matrices representation [26], [27] which typically reduces the computation cost associated with dense matrix to $O(N(\log N)^p)$. Here, the value of p is expected to be 1 in the cases of H-matrix construction and H-matrix-vector multiplication. Our preliminary study on the application of H-matrices to our problem was published in [28], in which a promising performance of H-matrices representation and possibility of large-scale analysis with over one million DOFs were successfully demonstrated. In this section, we present more detailed performance evaluations of our software, considering Newton–Raphson method and AMG preconditioning described in the previous sections.

Although the H-matrices representation of the coefficient matrices can be constructed directly, a different approach was adopted here. Matrix $A(\sigma)$ in Equation (10) varies in different nonlinear iteration steps such that the coefficient matrix itself also varies in each nonlinear iteration step. Thus, we generated and used the H-matrix representation of B, which was constant throughout the entire time-marching analysis, because the computation cost for constructing an H-matrix representation was generally high. For implementation, HACApK library [29]–[31], which was developed by some of the authors and provides an efficient parallel implementation of H-matrices representation for a distributed memory parallel computer, was used.

When the Kim's model given by Equation (6) is adopted, the dense matrices used to calculate $B_\perp$ from $\boldsymbol{J}$ need to be addressed. Since these matrices can be also regarded as discretized integral operators, HACApK library can efficiently handle them. The related computation costs are small in comparison with the computation costs of the coefficient matrix, as the former is used only once per each nonlinear iteration step while the latter is used in each linear iteration step. Hence, we primarily focus on the computation cost of the coefficient matrix in the performance evaluations below.

5.2. Computational performance using HACApK

5.2.1 Memory consumption

The numerical experiments described in this section were executed on host kf01, except for Section 5.2.3. The parallel performance described in Section 5.2.3 was measured on host kg01, which has a better memory bandwidth.

The amount of memory required by the H-matrix representation of B was evaluated in comparison with the direct representation by a two-dimensional array. Figure 10 shows the three test models prepared for the performance evaluation. The first model is a single straight conductor in which the vector integral term in Equation (2) is reduced to the scalar one. We adopted the model for a reference as the scalar integration is essentially suitable for the approximate cross approximation (ACA) algorithm [32], [33], which is used as a basic matrix approximation algorithm in the HACApK library. The second and third models are 1- and 50-turn single pancake coils, in which the vector integral in three dimensions must be addressed. The specifications of the test models are listed in Table 4.

Figure 11 shows the memory consumption versus number of DOFs with respect to the test problems. The solid and dashed lines represent the slopes of $O(N\log N)$ and $O(N^2)$, respectively. Since the amount of memory required by the direct representation of an $N \times N$ dense matrix is proportional to the square of N, the dashed line indicates an almost exact estimate of the memory consumption required by the direct representation. In contrast, the amount of memory consumed by the H-matrices representation in each test model can be approximately evaluated by $O(N\log N)$, which indicates the effectiveness of H-matrices representation for large-size problems.

The memory consumption in the 1-turn single pancake coil models shown in Figure 11 generally coincides with the memory consumption in single straight conductor models. This implies that the ACA

algorithm in the HACApK library may be effective for our applications that involve vector integral operation. Figure 11 also shows that, however, the performance of the H-matrices representation substantially deteriorates in the 50-turn pancake coil models. Our detailed analysis of the behavior of the HACApK library revealed that a specific situation in our application may cause an inappropriate decision with respect to the matrix admissibility [27] in the construction of H-matrix representation. More specifically, two clusters consisting of *adjacent* nodes that are geometrically close may reduce the performance of the HACApK library. Here, *adjacent* indicates that one or more triangle elements that simultaneously contain nodes in each of the two clusters exist in the computational mesh. When adjacent clusters consist of more than a few hundred nodes, a serious performance deterioration may occur. Figure 10(c) shows a considerably elongated and layered mesh where adjacent clusters consisting of more than a thousand nodes exist in the 50-turn pancake coil models. On the other hand, the clusters that are adjacent in the 1-turn single pancake coil models consist of a few dozen nodes at maximum, as shown in Figure 10(b). Thus a significant performance deterioration did not occur in the 1-turn models. This problem may be solved by modifying the admissibility criteria of the HACApK library.

5.2.2 Computation time and precision of matrix approximation

We examine the computation time and precision of matrix approximation in this sub-section, using H-matrices representation. The test models are stacked 50-turn single pancake coils, whose schematics are shown in Figures 2 and 3, and the specifications of the three models A–C with different numbers of DOFs are listed in Table 5. The top 50-turn single pancake coils in these models correspond to the ones in the 50-turn pancake coil models used to examine memory consumption. The current in all pancake coils was ramped up to 150 A in 30 s, with a time step of 1.5 s. Figure 12 shows the computation time consumed by the construction of the matrix, BiCGSTAB solver, and entire analyses. Because the increase of computational complexity of AMG preconditioning is sufficiently suppressed by the sparse matrix approximation, the most time-consuming part in the BiCGSTAB solver is matrix-vector multiplication, which has ordinarily $O(N^2)$ complexity. On the other hand, the number of matrix-vector multiplication function calls (the number of linear iteration) typically increases with increasing problem size. Hence, the dashed lines in Figure 12 give an exact or optimistic estimation of the computation time when direct representation of the dense matrix is used. However, the results based on H-matrices representation generally agree with the slope of $O(N\log N)$. These results demonstrate that computational performance of our analyses can be significantly improved using H-matrices representation and large-scale analyses not feasible in our past studies can be performed.

In our previous research [28], we showed that the simulation result of our analyses was barely affected by H-matrix representation. In this study, however, we examined the precision of the H-matrix approximation itself based on Frobenius norm. Table 6 shows the error between the dense matrix B directly computed and its H-matrix representation generated by the HACApK library. The error tolerance of the ACA algorithm in the HACApK library was set to the default value 10^{-3}. Thus, the Frobenius norm of the error is suppressed to be sufficiently small.

5.2.3 Parallel performance

We examined the parallel performance of our software incorporating the HACApK library. The computer used was host kg01 which produces a superior memory bandwidth in comparison with host kf01. Its specifications are listed in Table 1.

Figure 13 shows the speed-up rate in the construction of H-matrix representation, BiCGSTAB part, and entire analyses. Fairly good parallel performances which were not fully saturated in the range of measurement were achieved, except in the smallest test analysis. The parallel performance was relatively worse in the linear solver part in comparison with the construction of H-matrix representation, thereby limiting the performance of the entire analyses. The matrix-vector multiplication using H-matrices representation which is the computation kernel in the linear solver part had a relatively low computational density, which is defined by the ratio of the number of floating-point operations to memory references. Hence, the memory performance of the computers may have a substantial effect on the elapsed time. Figure 13(b) shows the plot of the normalized effective memory bandwidth of host kg01 with dashed lines, measured by using STREAM benchmark. The array size adopted in the benchmark was determined such that the amount of the memory use was equal to the memory required by the H-matrices representation in each test model. The behavior of the speed-up rate obtained in the largest test model was close to the effective memory bandwidth. Moreover, Figure 14 shows the comparison of the normalized memory bandwidths and speed-up rates of the linear solver part with respect to Model C on 3 different environments with 16 cores: all cores on kf01, on kg01, and 8 cores each on kf01 and kf02. The performance of analysis was strongly

correlated to the effective memory bandwidth. These results suggest that using computers with a better memory performance will further improve the parallel performance of our analyses.

6. Examples of analyses

In this section, at first, we present a validation of our software implementing the numerical methods above: we compare the ac loss and the lateral current density distribution calculated by using our novel electromagnetic field analysis model with those calculated by using pre-existing cross-sectional one-dimensional electromagnetic field analysis model [4], [34], [35] as well as the ac loss calculated by using Brandt's equation [36]. The analyzed object is a single straight coated conductor exposed to a sinusoidal external magnetic field B_{ext}, whose direction is normal to the wide face of the coated conductor and whose frequency is 50 Hz. The length and width of the coated conductor are 100 mm and 4 mm, respectively in our novel model, whereas the length is not considered in the pre-existing model and Brandt's equation. The critical current density J_c and n-value of the coated conductor are 2.5×10^{10} A m^{-2} and 30, respectively, and independent of magnetic field. Moreover, the thickness of the superconducting layer is 1 μm while the critical current I_c of the coated conductor is 100 A.

In Figure 15, we compare the ac losses per cycle per unit length of the coated conductor. The ac loss calculated by our novel model and that calculated by the pre-existing model show good agreement, even though the former is the averaged value in the 100-mm coated conductor. The slight deviation of the ac loss calculated by Brandt's equation from the others might be caused by the thin-strip approximation in our novel model and the pre-existing model. Note that Brandt's equation is not based on the thin-strip approximation. As shown in Figure 16, the lateral current density distribution at the center of the coated conductor calculated by our novel model agrees well with the lateral current density distribution calculated by the pre-existing model.

Moreover, some examples of analyses performed by using the software are given. All computations were carried out with Xeon E5 2667v4 (kf01, kf02) and/or Xeon Gold 6136 (kg01). Table 7 summarizes the analyses.

The first example is an analysis of a saddle-shaped coil of a cosine-theta dipole magnet for a rotating gantry for carbon cancer therapy [28]. The number of DOFs of the model was more than 1.5 million; the memory consumption by H-matrices was 177 GB, while that by a dense matrix had been 16.7 TB; the computation time was 78 h by 56 parallel processes. The current density distribution in coated conductors calculated from T distribution was visualized, and ac loss and influence of SCIF on the field quality of the magnet were evaluated. Figure 17 shows a temporal profile of current in the analysis. Figure 18 shows the lateral current density distributions at various time steps in a coated conductor of the saddle-shape coil shown in Table 7 column 2. The calculated time evolution of the influence of SCIF on dipole component of the magnetic field and the ac loss are shown in Figure 19.

The second example is an analysis of coils of a superferric magnet for a rapid cycling synchrotron [37]. The number of DOFs of the model was approximately 1.1 million; the memory consumption by H-matrices was 330 GB, while that by a dense matrix had been 9.29 TB. The computation time was 151 h by 56 parallel processes. This magnet was composed of coils wound with coated conductors and iron yoke. The influence of the nonlinear B–H characteristics of iron yoke on magnetic field distribution in the coated conductors was considered as B_{ext} in Equation (2). Ac losses in the coils were calculated as a result of the analysis. The calculated time evolution of the entire ac loss of the coils is shown in Figure 20 with temporal profile of current.

The third example is an analysis of single pancake coils wound with multifilament coated conductor [14], [19]. The analyzed sample pancake coil was exposed to an external cusp magnetic field generated by other coils. The spiral geometry of the analyzed pancake coil was considered, and the results were compared with experiments; the number of DOFs of the model was approximately 0.37 million; the memory consumption by H-matrices was 20.3 GB, while that by a dense matrix had been 1.02 TB. The computation time was 6–8 h by 16 parallel processes. Thus, the measured and calculated temporal evolutions of the magnetic field at the center of the coil are consistent with each other.

7. Conclusions

We developed a novel software for large-scale electromagnetic field analyses of coils wound with coated conductors based on current-vector-potential formulation with thin-strip approximation. We also carried out analyses of various coils wound with coated conductors, e.g., an analysis of a cosine-theta dipole magnet whose number of DOFs was more than 1.5 million. For performing such large-scale analyses, we combined

three numerical techniques in our software, which are: Newton–Raphson method, AMG preconditioning, and H-matrices representation.

The Newton–Raphson method was not used in our previous software because it leads to highly ill-conditioned linear systems of equations. However, the novel AMG preconditioning technique, which consists of the AMG method itself and sparse matrix approximation, can solve this problem. The computation time required to solve the equation with a strong nonlinear property of superconductor can be significantly reduced by applying AMG preconditioning to the linear-system solver in the Newton–Raphson method.

Meanwhile, AMG preconditioning is also effective in improving the convergence of the linear solvers when the conductivity of the analyzed object varies considerably, e.g., multifilament coated conductor. The new software enables us to perform a robust analysis of SCIF in a coil wound with multifilament coated conductor.

However, the main bottleneck which affected the scalability of our analyses was $O(N^2)$ computational complexity, due to the dense matrix property originating from current-vector-potential formulation. We successfully reduced the computational complexity of our analyses to approximately $O(N\log N)$, using HACApK library. Hence, the memory consumption and computation time can be drastically reduced by using H-matrices representation. Moreover, our software was quite compatible with parallel computation, using MPI library.

Furthermore, the electromagnetic field analysis of an HTS (High-Temperature Superconductor) coil with 1.5 million DOFs, a coil wound with multifilament coated conductor with 0.37 million DOFs, and coils of a superferric magnet for a rapid cycling synchrotron with 1.1 million DOFs were performed by combining the numerical techniques. Since the scalability and robustness of our analyses were significantly improved by the novel numerical techniques, larger-scale problems can be addressed by preparing a parallel computer system with more processors. The memory performance of the computer system will be important in such large-scale analyses, as well as the processing speed.

In addition, we plan to conduct simulations based on distributions of *E-J* characteristics of superconductors to make precise comparison between the simulations and measurements, using the developed software. In this case, the model size should be larger than it in the simulation using geometrical symmetry because the geometrical symmetry of coils wound with coated conductors was not established. Furthermore, we plan to enable simulations of coils wound with HTS cables, such as Roebel cable or CORC cable. Since HTS cables have three-dimensional geometry in their geometrical configuration, the simulations costs of coils wound with HTS cables is quite expensive. However, we believe that it will be able to conduct in future, using our software.

Acknowledgement

This work was supported in part by the Japan Society for the Promotion of Science with Grant Number KAKENHI 16H02326, the Japan Science and Technology Agency under the S-Innovation Program, and the Ministry of Education, Culture, Sports, Science, and Technology of Japan under the Innovative Nuclear Research and Development Program.

References

[1] Amemiya N, Murasawa S, Banno N and Miyamoto K 1998 Numerical modelings of superconducting wires for AC loss calculations *Phys. C.* **310** 16–29
[2] Nii M, Amemiya N and Nakamura T 2011 Three-dimensional model for numerical electromagnetic field analyses of coated conductors and its application to Roebel cables *Supercond. Sci. Technol.* **25** 095011
[3] Sogabe Y, Tsukamoto T, Mifune T, Nakamura T and Amemiya N 2015 Efficient and practical models for numerical electromagnetic field analyses of three-dimensional-shape coils wound with coated conductor *IEEE Trans. Appl. Supercond.* **25** 4900205
[4] Amemiya N and Akachi K 2008 Magnetic field generated by shielding current in high T_c superconducting coils for NMR magnets *Supercond. Sci. Technol.* **21** 095001
[5] Amemiya N, Sogabe Y, Sakashita M, Iwata Y, Noda K, Ogitsu T, Ishii Y and Kurusu T 2016 Magnetisation and field quality of a cosine-theta dipole magnet wound with coated conductors for rotating gantry for hadron cancer therapy *Supercond. Sci. Technol.* **29** 024006
[6] Sogabe Y, Sakashita M, Nakamura T, Ogitsu T and Amemiya N 2016 Influence of magnetization on field quality in cosine-theta and block design dipole magnets wound with coated conductors *Supercond. Sci. Technol.* **29** 045012
[7] Noguchi S, Monma K, Igarashi H and Ishiyama A 2016 Investigation of current flow between turns of NI REBCO pancake coil by 2-D finite-element method *IEEE Trans. Appl. Supercond.* **26** 4901205
[8] Pardo E 2016 Modeling of screening currents in coated conductor magnets containing up to 40000 turns *Supercond. Sci. Technol.* **29** 085004
[9] Grilli F, Zermeño V M R and Takayasu M 2015 Numerical modeling of twisted stacked tape cables for magnet applications *Phys. C.* **518** 122–5
[10] Zermeño V M R and Grilli F 2014 3D modeling and simulation of 2G HTS stacks and coils *Supercond. Sci. Technol.* **27** 044025
[11] Ueda H, Fukuda M, Hatanaka K, Wang T and Ishiyama A 2013 Spatial and temporal behavior of magnetic field distribution due to shielding current in HTS coil for cyclotron application *IEEE Trans. Appl. Supercond.* **23** 4100805
[12] Ueda H, Imaichi Y, Wang T, Ishiyama A, Noguchi S, Iwai S, Miyazaki H, Tosaka T, Nomura S, Kurusu T, Urayama S and Fukuyama H 2016 Numerical simulation on magnetic field generated by screening current in 10-T-class REBCO coil *IEEE Trans. Appl. Supercond.* **26** 4701205
[13] van Lanen E P A, van Nugteren J and Nijhuis A 2012 Full-scale calculation of the coupling losses in ITER size cable-in-conduit conductors *Supercond. Sci. Technol.* **25** 025012
[14] Carr W J and Oberly C E 1999 Filamentary YBCO conductors for AC applications *IEEE Trans. Appl. Supercod.* **9** 1475–8
[15] Cheng H, Greengard L and Rokhlin V 1999 A fast adaptive multipole algorithm in three dimensions *J. Comput. Phys.* **155** 468–98
[16] Hamada S and Takuma T 2003 Effective preconditioning technique to solve a full linear system for the fast multipole method *IEEE Trans. Magn.* **39** 1666–9
[17] Amemiya N, Tsukamoto T, Nii M, Komeda T, Nakamura T and Jiang Z 2014 Alternating current loss characteristics of a Roebel cable consisting of coated conductors and a three-dimensional structure *Supercond. Sci. Technol.* **27** 035007
[18] Ichiki Y and Ohsaki H 2004 Numerical analysis of AC losses in YBCO coated conductor in external magnetic field *Physica C* **412–414** 1015–20
[19] Amemiya N, Tominaga N, Toyomoto R, Nishimoto T, Sogabe Y, Yamano S and Sakamoto H 2018 Coupling time constants of striated and copper-plated coated conductors and the potential of striation to reduce shielding-current-induced fields in pancake coils *Supercond. Sci. Technol.* **31** 025007
[20] Jin J 2014 *The finite element method in electromagnetics* New York, NY, USA: Wiley 11–93
[21] Binns J, Lawrenson P J and Trowbridge C W 1992 The Analytical and Numerical Solution of Electric and Magnetic Fields Chichester, U.K.: Wiley 231–255
[22] Amemiya N, Miyamoto K, Banno N and Tsukamoto O 1997 Numerical analysis of AC losses in high-Tc superconductors based on E–J characteristics represented with n-value *IEEE Trans. Appl. Supercond.* **7** 2110–3
[23] Kim Y, Hempstead C and Strnad A 1962 Critical persistent currents in hard superconductors *Phys. Rev. Lett.* **9** 306–9
[24] Saad Y 2003 *Iterative Methods for Sparse Linear Systems* (SIAM) 244–7, 455–64

[25] Stuben K 2000 Algebraic multigrid (AMG): an introduction with applications, Trottenberg U, Schuller A, and Oosterlee C eds. *Multigrid* Academic Press 413–532
[26] Hackbusch W 1999 A sparse matrix arithmetic based on H-matrices. Part I: Introduction to H-matrices *Computing* **62** 89–108
[27] Hackbusch W and Khoromskij B N 2000 A sparse H-matrix arithmetic. Part II: Application to multi-dimensional problems *Computing* **64** 21–47
[28] Tominaga N, Mifune T, Ida A, Sogabe Y, Iwashita T and Amemiya N 2018 Application of hierarchical matrices to large-scale electromagnetic field analyses of coils wound with coated conductors *IEEE Trans. Appl. Supercond.* **28** 4900305
[29] ppOpen-HPC Project, ppOpen-APPL/BEM 2014 http://ppopenhpc.cc.u-tokyo.ac.jp/
[30] Ida A, Iwashita T, Mifune T and Takahashi Y 2014 Parallel hierarchical matrices with adaptive cross approximation on symmetric multiprocessing clusters *J. Inf. Process.* **22** 642–50
[31] Ida A, Iwashita T, Ohtani M and Hirahara K 2015 Improvement of hierarchical matrices with adaptive cross approximation for large-scale simulation *J. Inf. Process* **23** 366–72
[32] Kurtz S, Rain O and Rjasanow S 2002 The adaptive cross-approximation technique for the 3-D boundary-element method *IEEE Trans. Magn.* **38** 421–4
[33] Bebendorf M and Rjasanow S 2003 Adaptive low-rank approximation of collocation matrices *Computing* **70** 1–24
[34] Sugita S and Ohasaki H 2003 FEM analysis of current limiting characteristics of a superconducting thin film current limiting device by the current vector potential method *IEEE Trans. Appl. Supercond.* **13** 2020–3
[35] Brambilla R, Grilli F, Martini L and Sirois F 2008 Integral equations for the current density in thin conductors and their solution by the finite-element method *Supercond. Sci. Technol.* **21** 105008
[36] Brandt E H and Indenbom M 1993 Type-II-superconductor strip with current in a perpendicular magnetic field *Phys. Rev. B* **48** 12893-906
[37] Sogabe Y, Yasunaga M, Fuwa Y, Kuriyama Y, Uesugi T, Ishi Y, and Amemiya N 2019 AC losses in HTS coils of superferric dipole and combined-function magnets for rapid-cycling synchrotrons *IEEE Trans. Appl. Supercond.* **29** 5900505

Table 1. Specifications of computers used in the numerical experiments.

Host name	Processors	Memory	Theoretical memory bandwidth
kf01[*1]	Xeon E5 2667v4 (Broadwell, 3.20 GHz, 8 core) 2 CPU	512 GB	153.6 GB/s
kf02[*1]	Xeon E5 2667v4 (Broadwell, 3.20 GHz, 8 core) 2 CPU	512 GB	153.6 GB/s
kg01[*1]	Xeon Gold 6136 (Skylake, 3.00 GHz, 12 core) 2 CPU	384 GB	250 GB/s

[*1] Intel Fortran Compiler 2018 with –O3 option and Intel MPI library 2018 were used to compile our software.

Table 2. Specifications of stacked single pancake coils wound with monofilament coated conductors in the test model to examine the effectiveness of AMG preconditioning.

Number of stacked pancake coils	20
Analyzed object	Top single pancake coil (The other 19 single pancake coils were not analyzed. Uniform current density was assumed for the pancake coils and treated as source of B_{ext} in Equation (2).)
Inner radius	20 mm
Number of turns	10
Conductor width	4 mm
Conductor thickness	0.1 mm
Separation between turns	0.1 mm
Separation between coils	0.1 mm
Thickness of superconductor layer	1 μm
E_0 of E–J power law characteristic given by Equation (5) [*1]	1×10^{-4} V/m
n of E–J power law characteristic given by Equation (5) [*1]	30
Critical current density at zero field (J_{c0})	2.5×10^{10} A/m^2
Constant B_0 in Kim's model given by Equation (6) [*2]	∞
Current profile	Ramped up to 50 A in 10 s for all pancake coils
Time step	0.5 s
Meshes along 1 turn / meshes across width	30 / 20

[*1] Equation (5): $E = E_0(J/J_c(B_\perp))^n$.
[*2] Equation (6): $J_c(B_\perp) = J_{c0}[B_0/(B_0 + |B_\perp|)]$.

Table 3. Specifications of stacked single pancake coils wound with multifilament coated conductor in the test model to examine the effectiveness of AMG preconditioning.

Number of stacked pancake coils	20
Analyzed object	Top single pancake coil (The other 19 single pancake coils were not analyzed. One line current at the center of the coated conductor was assumed in the non-analyzed pancake coils and treated as source of B_{ext} in Equation (2).)
Inner radius	20 mm
Number of turns	10
Conductor width	4 mm
Conductor thickness	0.1 mm
Separation between turns	0.1 mm
Separation between coils	0.1 mm
Number of filaments	2
Groove between filaments	50 μm
Length of normal conductor region simulating copper terminals	10 mm
Thickness of superconductor layer	1 μm
E_0 of E–J power law characteristic given by Equation (5) [*1]	1×10^{-4} V/m
n of E–J power law characteristic given by Equation (5) [*1]	30
Critical current density at zero field (J_{c0})	2.5×10^{10} A/m^2
Constant B_0 in Kim's model given by Equation (6) [*2]	∞
Conductivity of groove	1×10^{9} S/m
Conductivity of current terminals	1×10^{13} S/m
Current profile	Ramped up to 50 A in 10 s for all pancake coils
Time step	0.5 s
Meshes along 1 turn / meshes across width	30 / 20 (SC) + 1 (normal groove)

[*1] Equation (5): $E = E_0(J/J_c(B_\perp))^n$.
[*2] Equation (6): $J_c(B_\perp) = J_{c0}[B_0/(B_0 + |B_\perp|)]$.

Table 4. Specifications of test models for the performance evaluation of memory use by H-matrix representation.

	Single straight conductor	1-turn single pancake coil	50-turn single pancake coil
Length/Inner radius	100 mm	50 mm	50 mm
Number of turns	N/A	1	50
Conductor width	4 mm	4 mm	4 mm
Conductor thickness	0.1 mm	0.1 mm	0.1 mm
Separation between turns	N/A	0.1 mm	0.1 mm
Thickness of superconductor layer	1 μm	1 μm	1 μm
Number of DOFs	2,761 / 10,521 / 41,041 / 162,081	651 / 2,121 / 31,521 / 105,021	31,521 / 52,521 / 105,021 / 315,021

Table 5. Specifications of stacked single pancake coils wound with monofilament coated conductors in the test model for the performance evaluation of computational time and precision using H-matrix representation.

Number of stacked pancake coils	30
Analyzed object	Top single pancake coil (The other 29 single pancake coils were not analyzed. Uniform current density was assumed for the pancake coils and treated as source of B_{ext} in Equation (2).)
Inner radius	50 mm
Number of turns	50
Conductor width	4 mm
Conductor thickness	0.1 mm
Separation between turns	0.1 mm
Separation between coils	0.1 mm
Thickness of superconductor layer	1 μm
E_0 of E–J power law characteristic given by Equation (5) [*1]	1×10^{-4} V/m
n of E–J power law characteristic given by Equation (5) [*1]	32
Critical current density at zero field (J_{c0})	1.91×10^{11} A/m^2
Constant B_0 in Kim's model given by Equation (6) [*2]	0.7 T
Current profile	Ramped up to 150 A in 30 s for all pancake coils
Time step	1.5 s
Number of DOFs	Model A: 31,521 Model B: 105,021 Model C: 315, 021

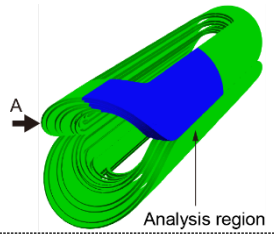
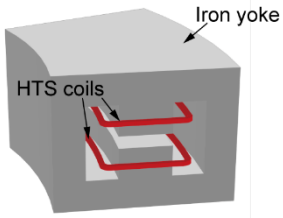
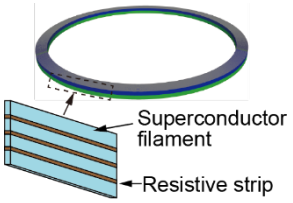
[*1] Equation (5): $E = E_0(J/J_c(B_\perp))^n$.
[*2] Equation (6): $J_c(B_\perp) = J_{c0}[B_0/(B_0 + |B_\perp|)]$.

Table 6. Error of H-matrix representation.

	Error tolerance of ACA	Relative error [*1]
Model A	1×10^{-3}	2.12×10^{-8}

[*1] $\|B - B_H\|_F / \|B\|_F$. B_H denotes the H-matrix representation of B while $\|\cdot\|_F$ denotes the Frobenius norm.

Table 7. Examples of analyses.

Applications	Saddle-shaped coil of a cosine-theta dipole magnet for a rotation gantry for carbon cancer therapy	Coils wound with coated conductors in a superferric magnet for a rapid cycling synchrotron	Single pancake coil wound with multifilament coated conductor
Image of analyzed object			
Purpose of analyses	Evaluation of ac loss and influence of SCIF on field quality under time-varying transport current based on the exact 3D geometry of the coil	Evaluation of ac loss in the HTS coils based on the influence of magnetized iron yoke as an external magnetic field	Evaluation of temporal evolution of SCIF
Number of DOFs	1.52 million	1.13 million	0.37 million
Memory consumption by H-matrices representation	177 GB	330 GB	20.3 GB
Memory consumption by direct representation	16.7 TB	9.29 TB	1.02 TB
Reduction ratio of memory consumption by H-matrices representation	98.9%	96.4%	98.0%
Computation time	78 h	151 h	6-8 h
Number of cores	56[*1]	56[*1]	16[*2]
Obtained or expected effects of analyses	This evaluation is necessary for the design of actual magnets for accelerator systems [28].	Calculated ac loss value will be used to determine operating conditions [37].	The analyzed and measured magnetic fields were compared, and their results were consistent with each other.

*1: Hosts kf01, kf02, and kg01 were used.
*2: Host kf01 was used.

Figure captions

Figure 1. Flowchart of time-marching analysis with nonlinear and linear iterations.

Figure 2. Schematic of the analysis region of the stacked single pancake coils wound with coated conductors for the test model.

Figure 3. Mesh used in the test analyses of stacked pancake coils wound with coated conductor to examine the effectiveness of AMG preconditioning.

Figure 4. Results of the test analyses of stacked pancake coils wound with monofilament coated conductor: (a) number of nonlinear iteration; (b) computation time when BiCGSTAB method with no preconditioning is used. In the test analyses, current was ramped up to 50 A in 10 s while the time step was 0.5 s.

Figure 5. Convergence of BiCGSTAB method with no preconditioning in successive substitution/Newton–Raphson iteration and with AMG preconditioning in Newton–Raphson iteration step at the 10th time step in the test analyses of stacked single pancake coils wound with monofilament coated conductor.

Figure 6. Comparison of computation time in the test analyses of stacked pancake coils wound with monofilament coated conductor: (a) 10-turn single pancake coil and (b) 20-turn single pancake coil.

Figure 7. Three-dimensional geometry of the single pancake coil wound with multifilament coated conductor and mesh used in test analyses.

Figure 8. Convergence of BiCGSTAB solver with and without AMG preconditioning in a nonlinear iteration step in the test analyses involving multifilament coated conductor.

Figure 9. Number of nonlinear iterations when using AMG preconditioning in the test analysis of the single pancake coil wound with multifilament coated conductor. In the test analyses, current was ramped up to 50 A in 10 s with a time step of 0.5 s.

Figure 10. Test models of (a) a straight conductor, (b) 1-turn single pancake coil, and (c) 50-turn single pancake coil for performance evaluation, using H-matrix representation.

Figure 11. Memory consumption required for matrix B in the test model, using H-matrix representation.

Figure 12. Computation time consumed by (a) the construction of the matrix, (b) BiCGSTAB solver, and (c) entire analyses in the test analyses of the stacked 50-turn single pancake coils.

Figure 13. Speed-up ratio in (a) the construction of the H-matrix representation, (b) BiCGSTAB solver, and (c) entire analyses in the test analyses of the stacked single pancake coils.

Figure 14. Comparison of (a) normalized memory consumption and (b) BiCGSTAB solver on hosts kf01, kg01, and combination of kf01 and kf02 in the test analyses of the stacked single pancake coils.

Figure 15. Comparison of calculated ac losses by our novel model, pre-existing model, and Brandt's equation.

Figure 16. Comparison of lateral current density distributions calculated by our novel model and the pre-existing model.

Figure 17. Temporal profile of current for the analysis of the saddle-shape coil of the cosine-theta dipole magnet for a rotating gantry for carbon cancer therapy.

Figure 18. Lateral current density distributions at various time steps in a coated conductor of the saddle-shape coil shown in Table 7 column 2.

Figure 19. Analysis result of the saddle-shape coil of the cosine-theta dipole magnet for a rotation gantry for a carbon cancer therapy: (a) temporal evolution of the influence of SCIF on the dipole component of the magnetic field and (b) temporal evolution of the entire ac loss of the coil.

Figure 20. Analysis result of the coils wound with coated conductors in a superferric magnet for a rapid cycling synchrotron: (a) temporal profile of current in the analysis, and (b) temporal evolution of the entire ac loss of the coils.

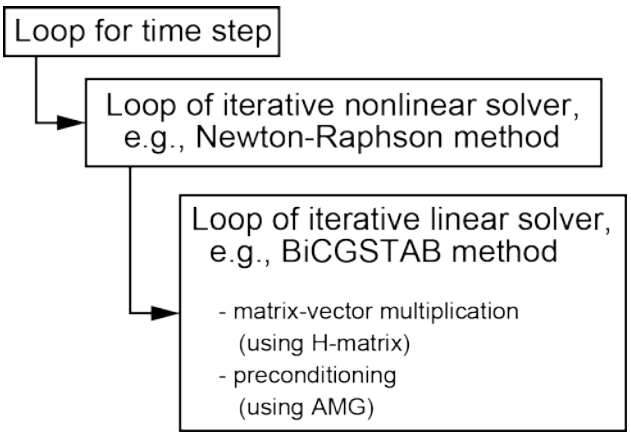


Figure 1.

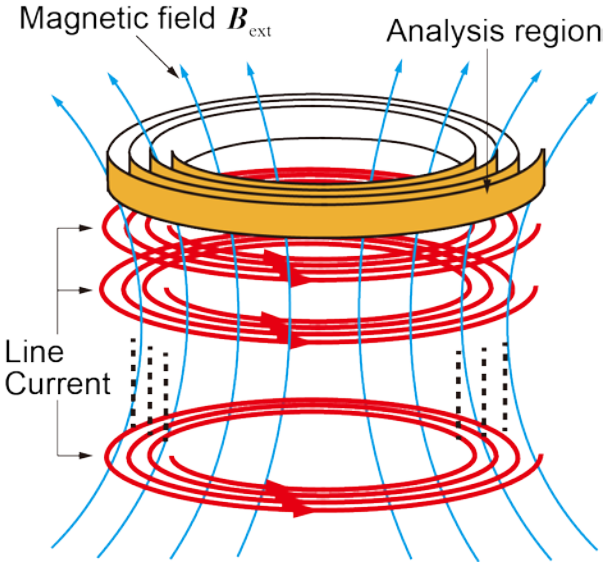


Figure 2.

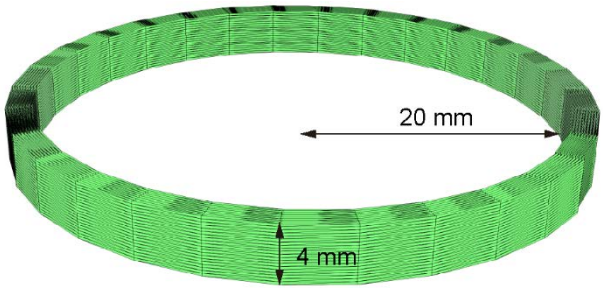


Figure 3.

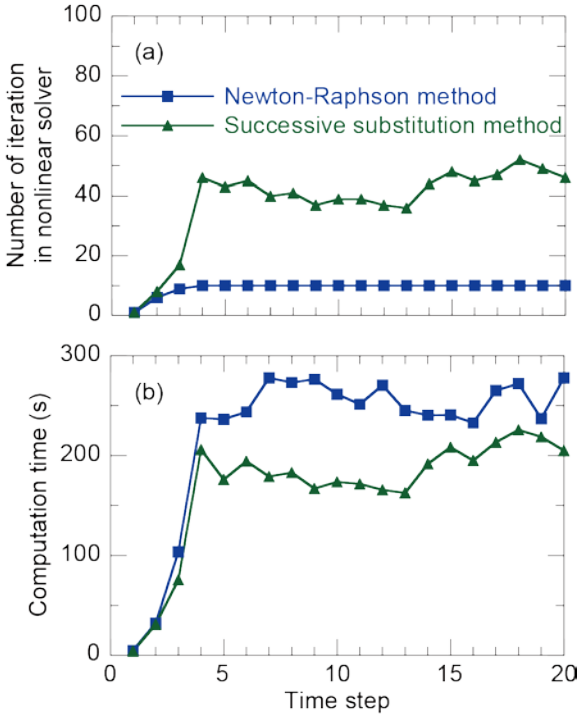


Figure 4.

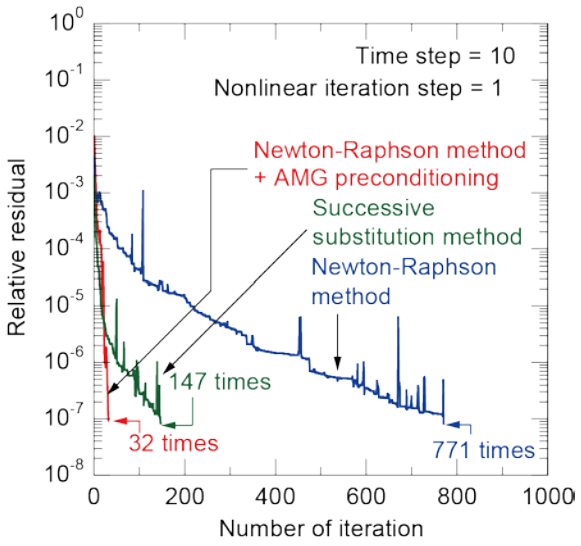


Figure 5.

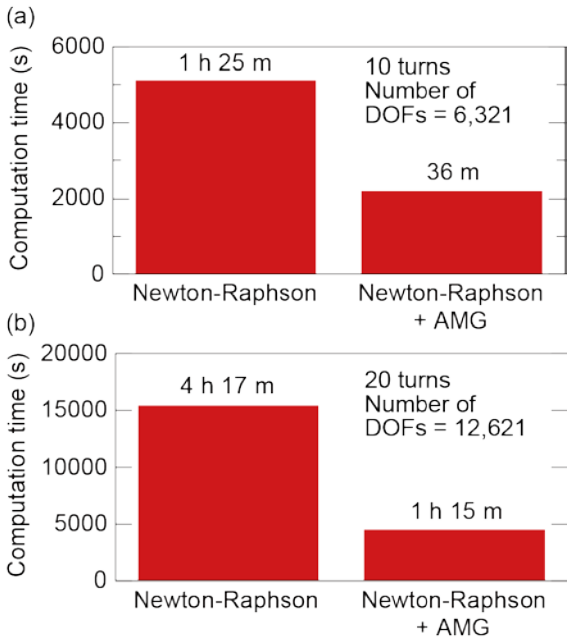


Figure 6.

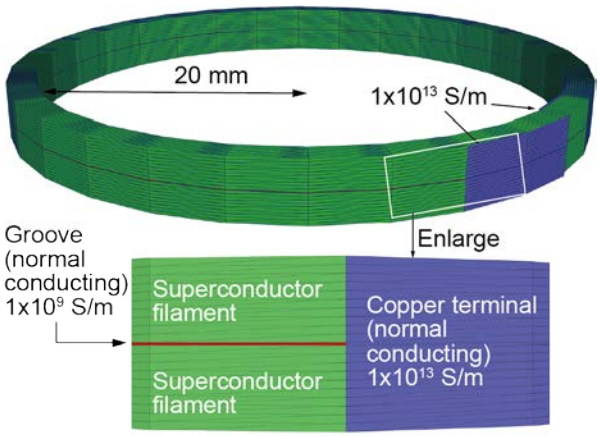


Figure 7.

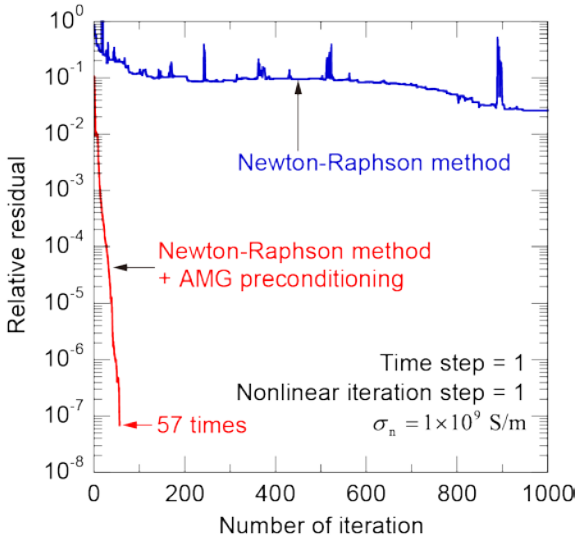


Figure 8.

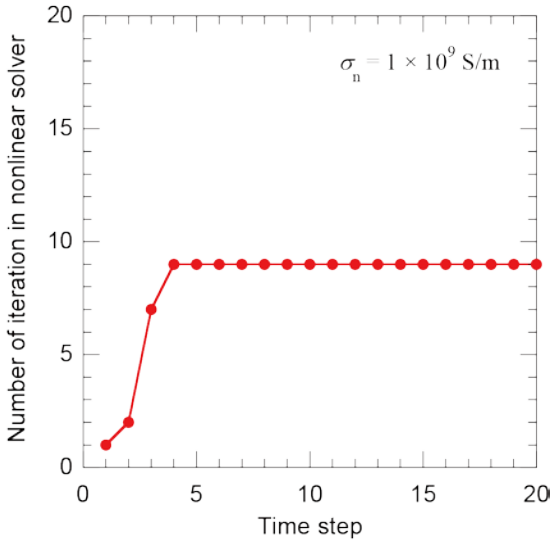


Figure 9.

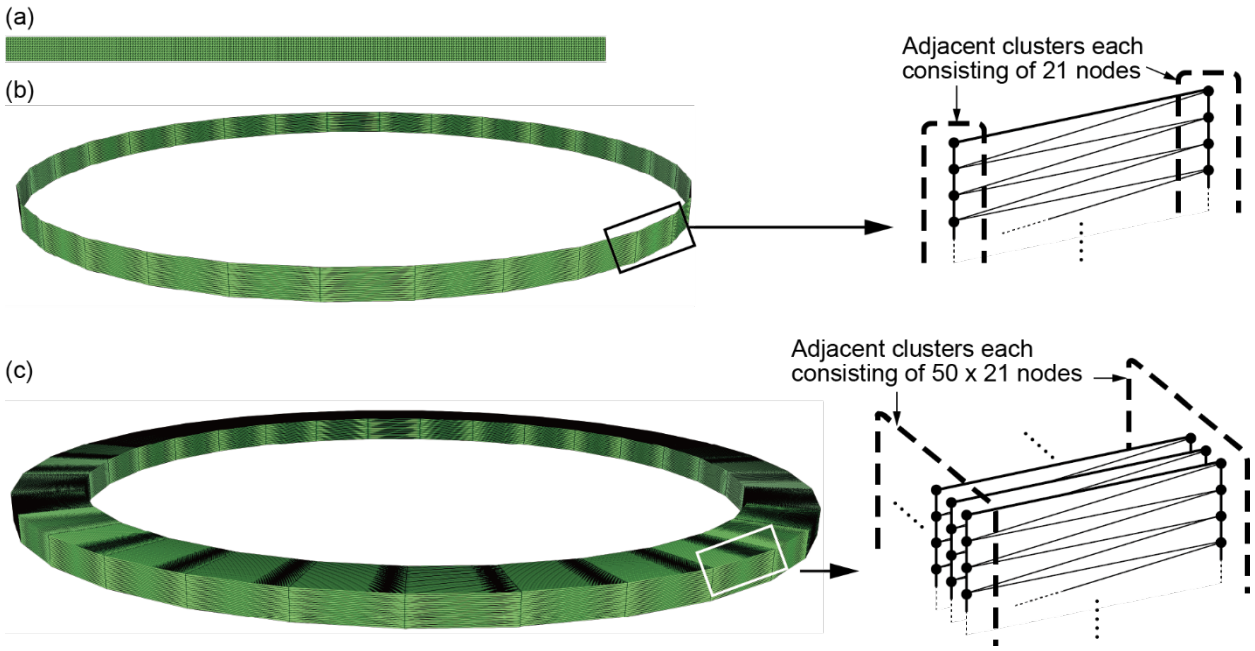


Figure 10.

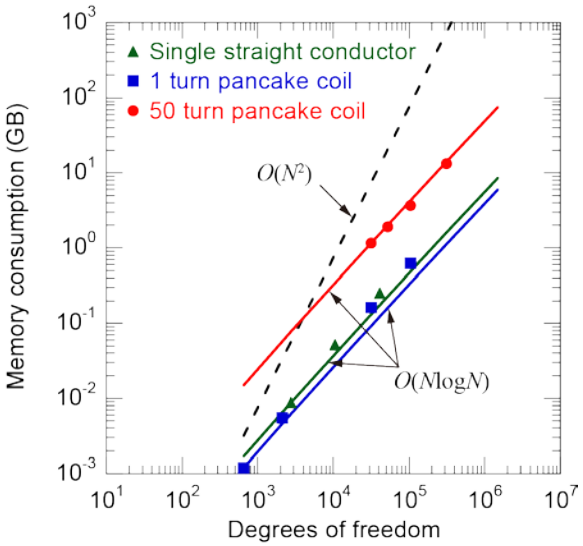


Figure 11.

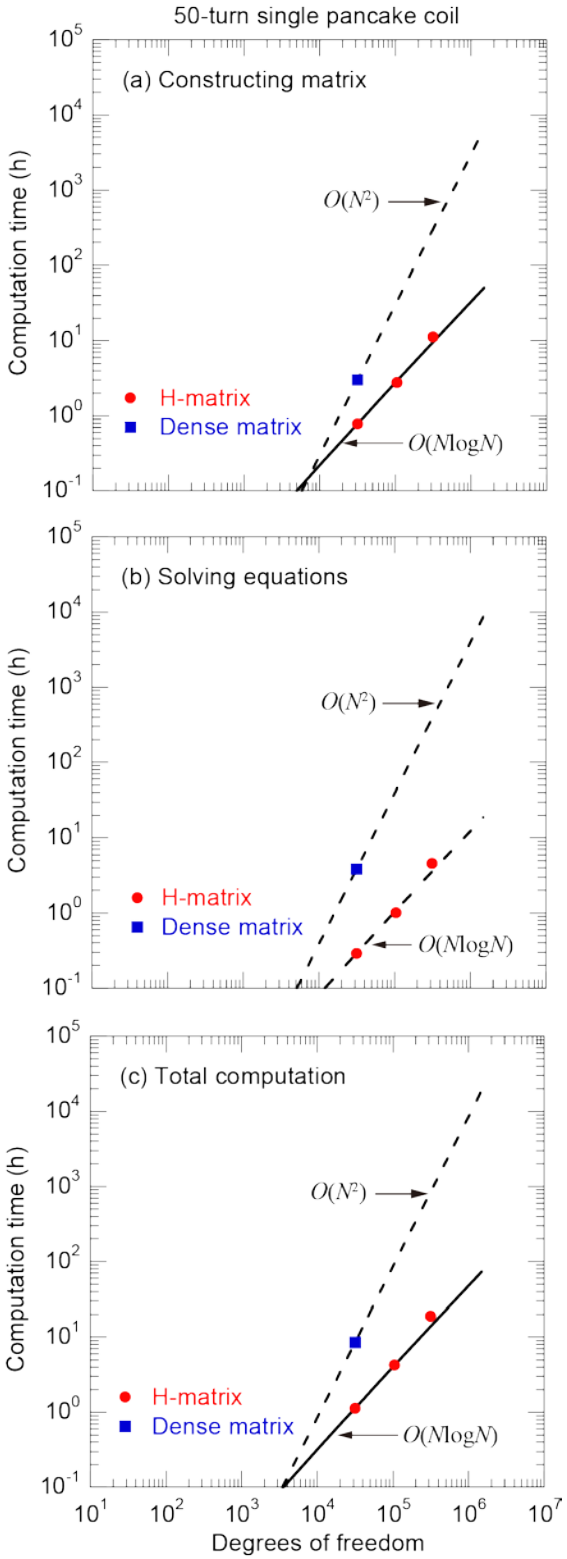


Figure 12.

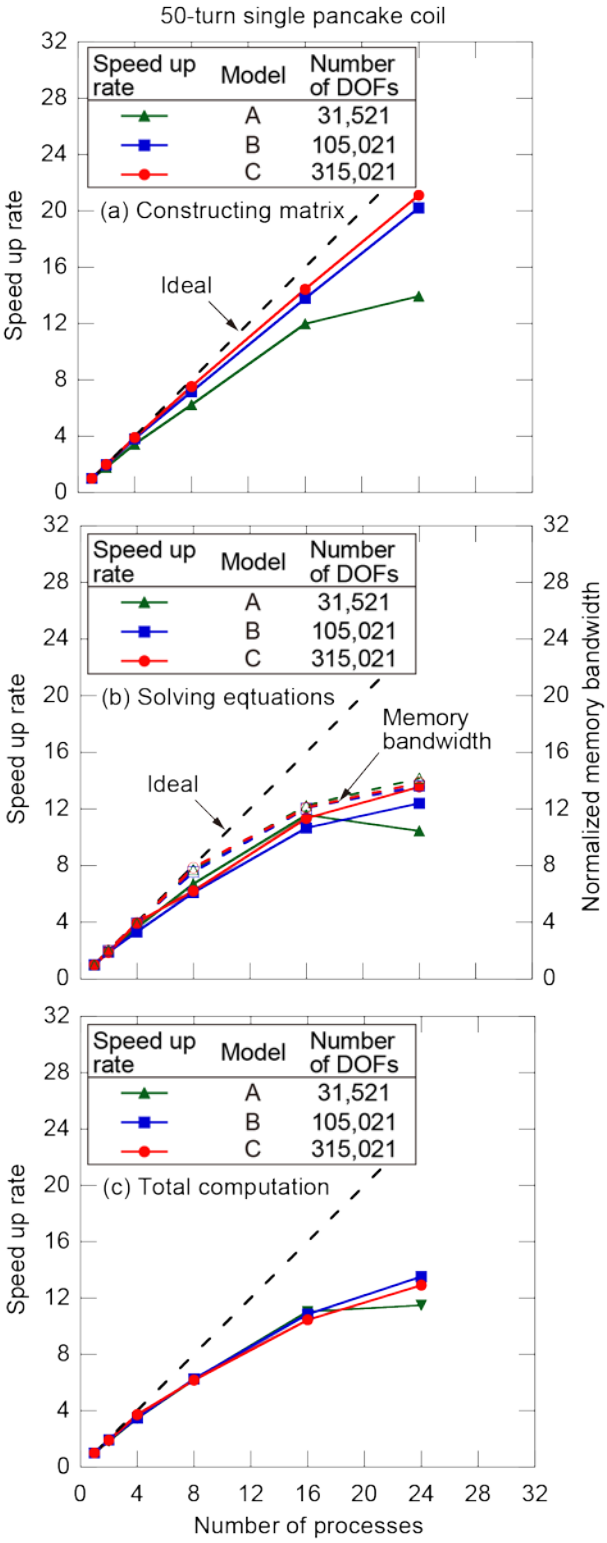


Figure 13.

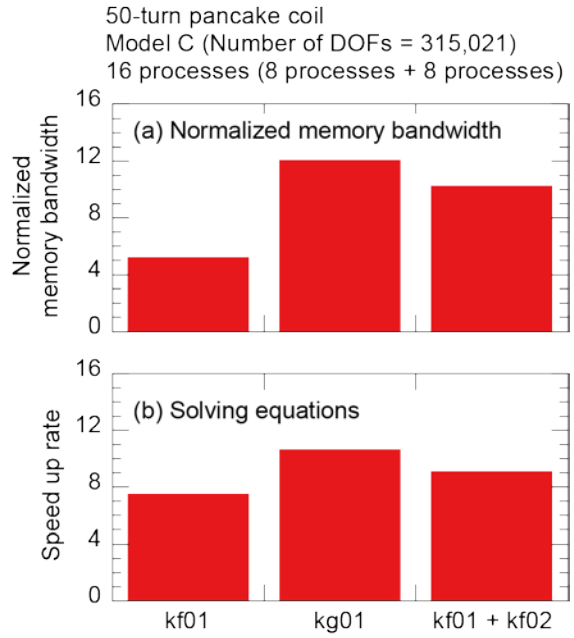


Figure 14.

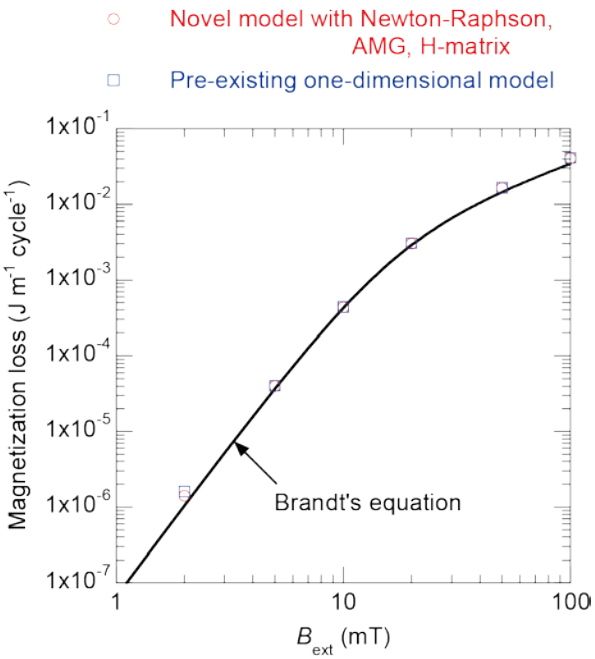


Figure 15.

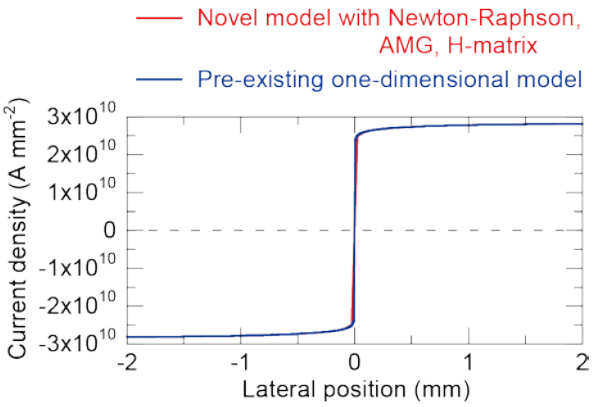


Figure 16.

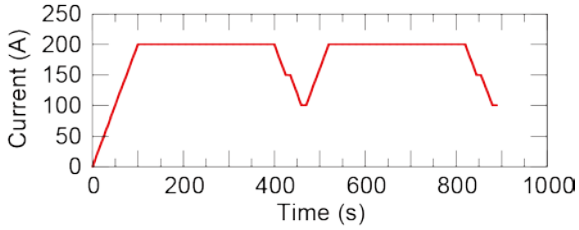


Figure 17.

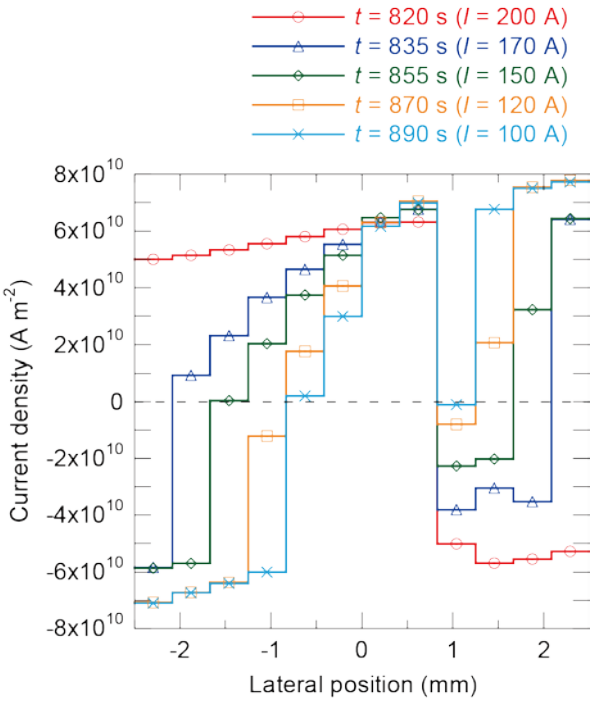


Figure 18.

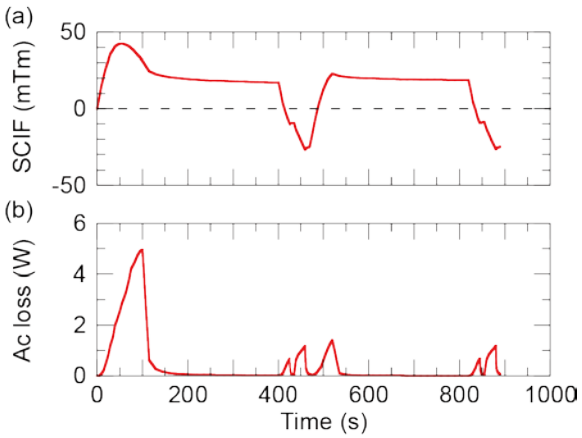


Figure 19.

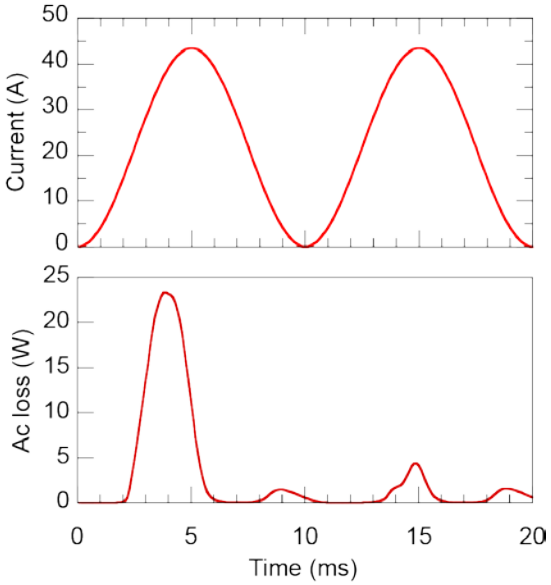


Figure 20.